



# On the Performance, Scalability and Sensitivity Analysis of a Large Air Pollution Model

Tzvetan Ostromsky<sup>1</sup>, Vassil Alexandrov<sup>2,3</sup>, Ivan Dimov<sup>1</sup>, and Zahari Zlatev<sup>4</sup>

<sup>1</sup> Institute of Information and Communication Technologies,  
Bulgarian Academy of Sciences,  
Acad. G. Bonchev, bl. 25A, 1113 Sofia, Bulgaria  
[ceco@parallel.bas.bg](mailto:ceco@parallel.bas.bg), [ivdimov@bas.bg](mailto:ivdimov@bas.bg)

<sup>2</sup> ICREA – Barcelona Supercomputing Centre (BSC-CNS),  
Carrer Jordi Girona 29, E-08034 Barcelona, Spain  
[vassil.alexandrov@bsc.es](mailto:vassil.alexandrov@bsc.es)

<sup>3</sup> Distinguished Visiting Professor in Computational Science,  
Technologico de Monterrey (ITESM), Campus Monterrey, Mexico

<sup>4</sup> National Centre for Environment and Energy, University of Århus,  
Frederiksborgvej 399 P.O. Box 358, DK-4000 Roskilde, Denmark  
[zz@dmu.dk](mailto:zz@dmu.dk)

## Abstract

Computationally efficient sensitivity analysis of a large-scale air pollution model is an important issue we focus on in this paper. Sensitivity studies play an important role for reliability analysis of the results of complex nonlinear models as those used in the air pollution modelling. There is a number of uncertainties in the input data sets, as well as in some internal coefficients, which determine the speed of the main chemical reactions in the chemical part of the model. These uncertainties are subject to our quantitative sensitivity study. Monte Carlo and quasi-Monte Carlo algorithms are used in this study.

A large number of numerical experiments with some special modifications of the model must be carried out in order to collect the necessary input data for the particular sensitivity study. For this purpose we created an efficient high performance implementation SA-DEM, based on the MPI version of the package UNI-DEM. A large number of numerical experiments were carried out with SA-DEM on the IBM MareNostrum III at BSC - Barcelona, helped us to identify a severe performance problem with an earlier version of the code and to resolve it successfully. The improved implementation appears to be quite efficient for that challenging computational problem, as our experiments show. Some numerical results with performance and scalability analysis of these results are presented in the paper.

**Keywords:** air pollution model, DEM, MPI, parallel algorithm, scalability, performance, speed-up, supercomputer, sensitivity analysis

# 1 Introduction to sensitivity analysis

In a popular definition (due to A. Saltelli [11]), sensitivity analysis (SA) is the study of how uncertainty in the output of a model can be apportioned to different sources of uncertainty in the model input. The uncertainties in the model input can be due to various reasons: inaccurate measurements or calculation, approximation, data compression, etc.. In order to measure the specific contribution of the uncertainty in each input parameter, considered to be a potential source, the sensitivity indices (SIs) have been introduced. Two kinds of sensitivity analysis have been discussed in the literature: local and global. Here we focus on global SA, which takes into account the whole domain of variation changes in the set of input parameters, and apportions the output uncertainty to the uncertainty in the input data. In particular, we search for efficient numerical algorithms for computing the global sensitivity indices (SIs) for the **Unified Danish Eulerian Model (UNI-DEM)** [16]. This large-scale model is described by a large system of partial differential equations (the number of equations being equal to the number of chemical species studied by the model). In dependence with the resolution of the discretization grid, the size of the final numerical problem can reach several millions of equations. This means that enormous computational tasks arise in the treatment of such a large-scale air pollution model, and great difficulties arise even when modern high-performance computers are used.

Therefore, it is highly desirable to simplify as much as possible the model. A careful sensitivity analysis is needed in order to decide where and how simplifications can be made. On the other hand, air pollution modelers might ask the extent to which their results depend on assumptions of initial conditions, boundary conditions, or chemical reaction rate constants. This analysis can give valuable information about how precise the model output is and identify which variables should be investigated more closely if the uncertainty is unacceptably high. The goals may be to rank the importance of input variables, improve precision, screening, and decision making.

Among quantitative methods, variance-based methods are the most often used [10]. The main idea of these methods is to evaluate how the variance of an input or a group of inputs contributes into the variance of an output variable. Variance-based methods deliver global, quantitative and model-independent sensitivity measures. A general sensitivity concept, namely the variance-based sensitivity analysis using a Monte Carlo technique, has been used in [6]. This concept is sampling-based, that is why a Monte Carlo simulation is applied. The techniques based on Monte-Carlo methods require a lot of simulations. The uncertain input parameters are modeled by random variables and characterized by their probabilistic density functions. The variance-based analysis focuses on the following questions: *"Which of the input variables variances influences the model output variance at most?"* and *"Which of the input variables has to be known more accurately to reduce the output variance?"*

The sensitivity analysis concept is used here to study the sensitivity of the output ozone concentrations, calculated by the Danish Eulerian Model, with respect to some perturbations in the input data or uncertain parameters. More specifically, we introduced two kinds of perturbations: to the values of the input anthropogenic emissions and to some chemical reactions rate coefficients. This study can be used for increasing the reliability of the model, as well as for identifying those parameters, which should be measured more precisely.

Let us consider a scalar function, representing some model output:  $u = f(\mathbf{x})$ , where the input parameters  $\mathbf{x} = (x_1, x_2, \dots, x_d) \in U^d \equiv [0, 1]^d$  are assumed to be independent (non-correlated) random variables with a known joint probability density function  $p(\mathbf{x}) = p(x_1, \dots, x_d)$ . In this way the output  $u$  becomes also a random variable (as it is a function of the random vector  $\mathbf{x}$ ) and let  $\mathbf{E}$  be its mathematical expectation. Let  $\mathbf{D}[\mathbf{E}(u|x_i)]$  be the variance of the conditional

expectation of  $u$  with respect to  $x_i$  and  $\mathbf{D}_u$  - the total variance according to  $u$ . This indicator is called *first-order sensitivity index* by Sobol [12] or sometimes *correlation ratio*.

*Total Sensitivity Index (TSI)* [12] of an input parameter  $x_i$ ,  $i \in \{1, \dots, d\}$  is the sum of the complete set of mutual sensitivity indices of any order (main effect, two-way interactions (second order), three-way interactions (third order), etc.):

$$S_{x_i}^{tot} = S_i + \sum_{l_1 \neq i} S_{il_1} + \sum_{l_1, l_2 \neq i, l_1 < l_2} S_{il_1 l_2} + \dots + S_{il_1 \dots l_{d-1}}, \quad (1)$$

where  $S_{il_1 \dots l_{j-1}}$  -  $j^{\text{th}}$  order sensitivity index for the parameter  $x_i$  ( $1 \leq j \leq d$ ),  $j = 1$  :  $S_i$  - the "main effect" of  $x_i$ . According to the values of their total sensitivity indices, the input parameters can be classified by the level of their importance. In most practical problems the high dimensional terms are sufficiently small and can be neglected, thus reducing significantly the number of summands in (1). In addition, a numerical approach for evaluating the small sensitivity indices that combines reduction of the mean value and correlated sampling suggested in [13] has been applied. Here we are not going into deeper details on the sensitivity analysis matter, because the main purpose of this paper is to discuss the performance and scalability of the numerical algorithms, used in this study. However, if the reader is particularly interested in that topic, in [2, 4, 5] he can find more detailed description, analysis and results of various sensitivity analysis studies performed so far.

## 2 The Danish Eulerian Model – description, versions, parallelization

The Danish Eulerian Model (DEM) [15] is a large-scale air pollution model, used to calculate the concentrations of various dangerous pollutants and other species over a large geographical region ( $4800 \times 4800$  km), covering the whole of Europe, the Mediterranean and some parts of Asia and Africa. It takes into account the main physical, chemical and photochemical processes between the studied species, the emissions, the quickly changing meteorological conditions.

The Danish Eulerian Model is mathematically represented as a large system of partial differential equations (2), in which the unknown concentrations of a large number of chemical species (pollutants and other chemically active components) take part. The main physical and chemical processes (advection, diffusion, chemical reactions, emissions and deposition) are represented as separate additive terms in the right-hand-side of this system.

$$\begin{aligned} \frac{\partial c_s}{\partial t} = & -\frac{\partial(uc_s)}{\partial x} - \frac{\partial(vc_s)}{\partial y} - \frac{\partial(wc_s)}{\partial z} + \\ & + \frac{\partial}{\partial x} \left( K_x \frac{\partial c_s}{\partial x} \right) + \frac{\partial}{\partial y} \left( K_y \frac{\partial c_s}{\partial y} \right) + \frac{\partial}{\partial z} \left( K_z \frac{\partial c_s}{\partial z} \right) + \\ & + E_s + Q_s(c_1, c_2, \dots, c_q) - (k_{1s} + k_{2s})c_s, \quad s = 1, 2, \dots, q. \end{aligned} \quad (2)$$

Here it is a short description of the functions, participating in the above system:

- $c_s = c_s(x, y, z, t)$  are the concentrations of the chemical species. All the functions described in the next items are also dependent on the space and time coordinates  $(x, y, z, t)$ , but for the sake of shortening the formulae they are usually not written explicitly.
- $u, v, w$  - the components of the wind vector  $\vec{W}(x, y, z, t)$  along the coordinate axes.

- $K_x, K_y, K_z$  – diffusion coefficients.
- $E_s$  – the emission functions.
- $k_{1s}, k_{2s}$  – dry / wet deposition coefficients.
- $Q_s(c_1, c_2, \dots, c_q)$  – non-linear functions describing the chemical reactions between species under consideration.

## 2.1 Numerical treatment of the model

The above large and rather complex system (2) is not suitable for direct numerical treatment. For the purpose of numerical solution it is split into submodels, which represent the main physical and chemical processes. The sequential splitting [7] is used in the production version of the model, although other splitting methods have also been considered and tested in some experimental versions [1].

Below the 3 basic submodels of DEM are given (obtained by using the most straightforward sequential splitting scheme in accordance with the major physical and chemical processes).

$$\frac{\partial c_s^{[1]}}{\partial t} = -\frac{\partial(uc_s^{[1]})}{\partial x} - \frac{\partial(vc_s^{[1]})}{\partial y} + \frac{\partial}{\partial x} \left( K_x \frac{\partial c_s^{[1]}}{\partial x} \right) + \frac{\partial}{\partial y} \left( K_y \frac{\partial c_s^{[1]}}{\partial y} \right) = A_1 c_s^{[1]}(t) \quad (3)$$

**(horizontal advection and diffusion)**

$$\frac{\partial c_s^{[2]}}{\partial t} = E_s + Q_s(c_1^{[2]}, c_2^{[2]}, \dots, c_q^{[2]}) - (k_{1s} + k_{2s})c_s^{[2]} = A_2 c_s^{[2]}(t) \quad (4)$$

**(chemistry, emission and deposition)**

$$\frac{\partial c_s^{[3]}}{\partial t} = -\frac{\partial(wc_s^{[3]})}{\partial z} + \frac{\partial}{\partial z} \left( K_z \frac{\partial c_s^{[3]}}{\partial z} \right) = A_3 c_s^{[3]}(t) \quad (5)$$

**(vertical transport)**

These submodels are solved numerically in a cycle at each time step  $t_i$ . The second and third submodel at  $i$ -th time step use for initial values of  $c_s^{[m]}(x, y, z, t_i)$ ,  $m = 2, 3$  the results  $c_s^{[m-1]}(x, y, z, t_i)$  of the previous submodel at the same time step, while the first submodel uses for initial values the results  $c_s^{[3]}(x, y, z, t_{i-1})$  of the third submodel at the previous time step  $t_{i-1}$ .

Spatial and time discretization makes each of the above submodels a huge computational task, challenging even for the most powerful supercomputers available nowadays. That is why the parallelization has always been a key point in the computer implementation of DEM since its very early stages. A coarse-grain parallelization strategy based on partitioning of the spatial domain appears to be the most efficient and well-balanced way on widest class of nowadays parallel machines (with not too many processors), although some restrictions apply.

In the submodel for the vertical transport – Finite Elements, followed by  $\theta$ -methods. We should mention that it is possible to switch off the vertical transport submodel (optionally), resulting in a simplified 2D version of the model with about 10 times less computational complexity, compared to the corresponding 3D version (the latter splits the atmosphere into 10 layers, in spite of the horizontal discretization step). The 2D version, although not so accurate, is quite practical when the solution time is critical, and especially in case of insufficient memory of the computing system to execute the 3D version.

## 2.2 UNI-DEM package, features and parameters

The development and improvements of DEM throughout the years has lead to a variety of different versions with respect to the grid-size/resolution, vertical layering (2D or 3D model respectively) and the number of species in the chemical scheme. The most prospective of them have been united in the package UNI-DEM. The available up-to-date versions, the selecting parameters and their optional values are shown in Table 1.

Table 1: User-determined parameters for selecting an appropriate UNI-DEM version

Parameter	Description	Supported optional values		
<b>NX = NY</b>	Grid size (Grid step)	$96 \times 96$ (50 km)	$288 \times 288$ (16.7 km)	$480 \times 480$ (10 km)
<b>NZ</b>	# layers (2D/3D)	1	10	
<b>NEQUAT</b>	# chemical species	35	56	168

A coarse-grain parallelization strategy based on partitioning of the spatial domain in strips or blocks is currently used in UNI-DEM. For the purpose of this study, the strip-based distributed memory parallelization of the model via MPI is used [3, 9]. It is based on partitioning of the horizontal grid, which implies certain restrictions on the number of MPI tasks and requires communication on each time step. Improving the data locality for more efficient cache utilization is achieved by using **chunks** to group properly the small tasks in the chemistry-deposition and vertical exchange stages. Additional pre-processing and post-processing stages are needed for scattering the input data and gathering the results, causing some overhead.

## 2.3 SA-DEM and its parallelization properties

A special parallel version (SA-DEM) of the UNI-DEM has been created to produce the necessary data for our sensitivity analysis algorithm (see [2, 8] for more detail). It has been implemented first on the IBM Blue Gene/P platform at the Bulgarian Supercomputing Center.

SA-DEM is composed of the following three parts:

- (i) A modification of UNI-DEM with ability to modify certain parameters, subject to SA study. By now we have been interested in some chemical rate constants as well as in the input data for the anthropogenic emissions. A small number of input parameters is reserved for this purpose.
- (ii) A driver routine that automatically generates a set of tasks to produce the necessary results for a particular SA study. It allows to perform in parallel a large number of runs with common input data (reusing it), producing at once a whole set of values on a regular mesh (used later for calculating the sensitivity indices).
- (iii) An additional program for extracting the necessary mean monthly concentrations and computing the normalised ratios to be analysed.

Significant improvements of the first version of SA-DEM were made by introducing two additional levels of parallelism: top-level(MPI) and bottom-level(OpenMP). They allow us to use efficiently the computational power of a typical state-of-the-art supercomputer cluster with multicore nodes.

A new subroutine that splits the global communicator `MPI_COMM_WORLD` and defines separate communicators for each of the top-level parallel tasks is introduced in SA-DEM. The communicators are very useful on the lower level of parallelism (where intensive communications are performed on each time-step).

### 3 Numerical results from experiments with SA-DEM on an IBM MareNostrum III supercomputer

In this section we present some scalability results (including execution times, speed-ups and parallel efficiencies), obtained from our experiments with SA-DEM on the largest supercomputer system in Spain - IBM MareNostrum III (managed by BSC - Barcelona). It consists of 3028 nodes IBM dx360 M4 (16 core) with 32 GB RAM per node. The total number of cores is 48488 (Intel SandyBridge-EP E5-2670, 2600 MHz). It has total RAM more than 94 TB, total disk storage about 1,9 PB and two interconnection networks (Infiniband / Gigabit Ethernet). The system has theoretical peak performance  $\sim 1$  PFLOPS. Table 2 contains results of experiments on different number of processors, showing the scalability of SA-DEM on the IBM MareNostrum III machine at BSC.

Table 2: Time (T) in seconds and speed-up **Sp** of SA-DEM (finest grid) supercomputer IBM MareNostrum III at BSC, Barcelona

Time and <b>speed-up</b> of SA-DEM (MPI only) on IBM MareNostrum III ( $480 \times 480 \times 1$ ) grid, 35 species, CHUNKSIZE=16								
# CPU	# nodes	Advection		Chemistry		TOTAL		
		T [s]	<b>Sp</b>	T [s]	<b>Sp</b>	T [s]	<b>Sp</b>	<i>E</i> [%]
10	1	83460	<b>10</b>	77273	<b>10</b>	171707	<b>10</b>	100 %
40	3	19448	<b>43</b>	16946	<b>46</b>	40471	<b>42</b>	106 %
80	5	9874	<b>85</b>	9047	<b>85</b>	22261	<b>77</b>	96 %
160	10	5250	<b>159</b>	4562	<b>169</b>	12875	<b>133</b>	83 %
320	20	2895	<b>288</b>	2403	<b>322</b>	8233	<b>209</b>	65 %
640	40	1522	<b>548</b>	1269	<b>609</b>	5387	<b>319</b>	50 %
960	60	1215	<b>687</b>	822	<b>940</b>	4075	<b>421</b>	44 %
1600	100	873	<b>956</b>	502	<b>1538</b>	3289	<b>522</b>	33 %

### 4 Improved data management parallel algorithm

In order to resolve the severe data transfer problem and to improve the total efficiency of the SA-DEM, the following changes were made in the strategy for data transfer and management. A small (fixed) number of processors are used only for I/O procedures (global file transfer and scatter/gather operations), as well as for exchange of data (via MPI) with the rest of the processors. In particular, 11 processors are reserved for the 11 meteorological input data sets and 5 - for the output data sets, (16 in total). The rest of the processors are executing the basic algorithm without the above-mentioned (overhead-causing) operations. Receiving and sending of (local) I/O data is done via MPI instead of reading/writing temporary files (normally, much faster on most modern supercomputers). The number of opened files in a time is kept constant

(independent of the number of MPI processes). Part of the non-scalable overhead (producing and using local temporary files) is fully avoided.

Table 3: Time (T) in seconds and speed-up **Sp** of SA-DEM (finest grid) with the improved data management strategy on the Spanish supercomputer IBM MareNostrum III at BSC, Barcelona

Time and <b>speed-up</b> of SA-DEM (MPI only) on IBM MareNostrum III ( $480 \times 480 \times 1$ ) grid, 35 species, CHUNKSIZE=16								
# CPU	# nodes	Advection		Chemistry		TOTAL		
		T [s]	<b>Sp</b>	T [s]	<b>Sp</b>	T [s]	<b>Sp</b>	<i>E</i> [%]
10	1	83460	<b>10</b>	77273	<b>10.00</b>	169015	<b>10</b>	100 %
56	4	20117	<b>41</b>	16695	<b>46</b>	37983	<b>44</b>	79 %
96	6	10778	<b>77</b>	9148	<b>84</b>	21286	<b>79</b>	83 %
176	11	5219	<b>160</b>	4409	<b>175</b>	11080	<b>153</b>	87 %
336	21	2862	<b>292</b>	2214	<b>349</b>	6567	<b>257</b>	77 %
656	41	1491	<b>560</b>	1162	<b>665</b>	3833	<b>441</b>	67 %
976	61	1197	<b>697</b>	772	<b>1001</b>	2814	<b>601</b>	62 %
1616	101	848	<b>984</b>	479	<b>1614</b>	2173	<b>778</b>	48 %

Table 3 contains results of scalability experiments with the improved data management version of SA-DEM (as described above) on the IBM MareNostrum III machine at Barcelona are presented. As in 2, the times are for one year modelling period. The first row of the table (for 10 CPU) is taken from Table 2, as we cannot apply the new parallelization strategy on 16 CPU or less. Even on 2 - 3 times more CPUs, the new algorithm will not work efficiently, as relatively large part of the CPUs will be used to deal with the communication and data distribution overhead only (not so heavy when the number of parallel tasks is relatively small), in contrast with the rest of the CPUs, which will be rather busy with the main computational work. The effect of this relatively poor load-balance can still be seen on the second row of the table (where 40 out of 56 CPUs are engaged in executing the main computational work).

With increasing the total number of CPUs, however, this effect disappears and we obtain better results for the total speed-up and efficiency (in comparison with those for the basic version, shown in Table 2). The tiny part of the processors, selected to deal with the I/O transfer and scatter/gather operations, helps a lot to avoid the communication bottleneck and to provide conditions for smooth and efficient work for the rest (now – the vast majority) of the CPUs executing in parallel the computational tasks. For the highest levels of parallelism the new parallel algorithm appears to be much more efficient.

## 5 Conclusions and plans for future work

From the results, obtained in our scalability experiments and shown above in this paper, the following conclusions can be drawn:

(i) SA-DEM is an efficient high performance tool, used to produce sensitivity analysis data for the DEM. Its parallel implementations on the most powerful supercomputer in Spain, IBM MareNostrum III, show very good scalability.

(ii) Chemistry, the most computationally expensive stage of DEM, has almost linear speed-up in the whole range of experiments.

(iii) Advection stage scales pretty well in most of the experiments, with certain slow-down in the highly parallel experiments (expected). It is caused by the significant boundary overlapping of the domain partitioning when approaching the inherent partitioning limitations.

(iv) With increasing the number of processors the time for I/O operations (non-scalable) becomes strongly dominant. As a consequence, the total efficiency decreases with increasing the CPU number (especially above certain high level of parallelizm in dependence with the grid size).

(v) The above effect has been significantly reduced by improving the data management strategy. It reduces the amount of I/O operations and executes part of them in parallel with the computations on the other stages, a kind of pipelining between the I/O-intensive part and the computationally-intensive stages of the model. The new data management strategy improves the overall speed-up and efficiency of DEM and SA-DEM, as shown by experiments on the IBM MareNostrum III supercomputer in BSC - Barcelona. These developments will allow us to use widely the most detailed and, respectively, the most computationally expensive finest-grid version in our future sensitivity studies of DEM.

## Acknowledgments

This research is supported in parts by the Bulgarian NSF Grant Efficient Parallel Algorithms for Large Scale Computational Problems (DFNI # I 02/20). Barcelona Supercomputing Centre (BSC) is kindly acknowledged too for granting us access and computer time for experiments on their most powerful supercomputers.

## References

- [1] I. Dimov, I. Faragó, Á. Havasi, Z. Zlatev, Operator splitting and commutativity analysis in the Danish Eulerian Model, *Math. Comp. Sim.* **67** (2004), pp. 217–233.
- [2] I. Dimov, R. Georgieva, S. Ivanovska, Tz. Ostromsky, Z. Zlatev, Studying the Sensitivity of the Pollutants Concentrations Caused by Variations of Chemical Rates, *Journal of Computational and Applied Mathematics Vol. 235, No. 2* (2010), pp. 391–402.
- [3] I. Dimov, K. Georgiev, Tz. Ostromsky, Z. Zlatev, Computational challenges in the numerical treatment of large air pollution models, *Ecological Modelling* **179**, Elsevier (2004), pp. 187–203.
- [4] I. Dimov, K. Georgiev, Tz. Ostromsky, Z. Zlatev, Sensitivity studies of pollutant concentrations calculated by UNI-DEM with respect to input emissions, *Central Europe Journal of Mathematics* **Vol. 11, No 8** (2013), pp. 1531–1545.
- [5] I. Dimov, R. Georgieva, Tz. Ostromsky, Monte Carlo Sensitivity Analysis of an Eulerian Large-scale Air Pollution Model, *Reliability Engineering and System Safety* **107**, Elsevier (2012), pp. 23–28.
- [6] I. Dimov, Z. Zlatev, Testing the sensitivity of air pollution levels to variations of some chemical rate constants, *Notes on Numerical Fluid Mechanics* **62** (1997) 167–175.
- [7] G. I. Marchuk, Mathematical modeling for the problem of the environment, *Studies in Mathematics and Applications*, No. 16, North-Holland, Amsterdam, (1985).
- [8] Tz. Ostromsky, I. Dimov, R. Georgieva, Z. Zlatev, Sensitivity Analysis of a Large-scale Air Pollution Model: Numerical Aspects and a Highly Parallel Implementation, In: *Large-Scale Scientific Computations*, LNCS-5910 (2010), Springer, pp. 197–205.
- [9] Tz. Ostromsky, Z. Zlatev, Parallel Implementation of a Large-scale 3-D Air Pollution Model, *Large Scale Scientific Computing* (S. Margenov, J. Wasniewski, P. Yalamov, Eds.), LNCS-2179, Springer (2001), pp. 309–316.



- [10] A. Saltelli, K. Chan, M. Scott, Sensitivity Analysis, Probability and Statistics series, John Wiley & Sons (2000).
- [11] A. Saltelli, S. Tarantola, F. Campolongo, M. Ratto, Sensitivity Analysis in Practice: A Guide to Assessing Scientific Models, Halsted Press New York (2004).
- [12] I.M. Sobol, Sensitivity estimates for nonlinear mathematical models, *Mathematical Modeling and Computational Experiment* **1** (1993), pp. 407–414.
- [13] I.M. Sobol, E. Myshetskaya, Monte Carlo Estimators for Small Sensitivity Indices, *Monte Carlo Methods and Applications* **13** (5-6) (2007) pp. 455–465.
- [14] WEB-site of the Danish Eulerian Model, available at:  
*<http://www.dmu.dk/AtmosphericEnvironment/DEM>*
- [15] Z. Zlatev, Computer treatment of large air pollution models, Kluwer, 1995.
- [16] Z. Zlatev, I. Dimov, Computational and Numerical Challenges in Environmental Modelling, Elsevier, Amsterdam (2006).